Note

Ce TP est à rendre selon le deadline fixé par votre enseignant.

Objectifs :

Comprendre l'Apprentissage Supervisé à travers la construction des Arbres de Décisions

I. Construction d'un Arbre de Décision pour un « Jeu de Tennis» avec R

Source : http://www.grappa.univ-lille3.fr/~ppreux/ensg/miashs/fouilleDeDonneesI/tp/arbres-de-decision/

Objectif : Construire un Arbre de Décision à partir de données climatiques, afin de prédire si on pourra jouer au Tennis ou non.

1. Chargement de la bibliothèque

Pour pouvoir construire des arbres de décision, on va utiliser la bibliothèque « **rpart** **»** de l'environnement R.

Il faut tout d'abord la rendre accessible. Pour cela, on tape la commande suivante :

```
library(rpart)
```

2. Importation de données

On commence par charger le jeu de données « Tennis1.txt ». Pour ce faire, placer cet entrepôt de données dans un data frame 'Tennis' :

```
1  #**********************************DATA_TENNIS**********************************
2  #*****Comprehension des donnees*****************************************************
3  tennis<-read.table(file = file.choose(),header = TRUE,fileEncoding = 'latin1')
4  View(tennis)
5  tennis
```

3. Construction et Visualisation de l'arbre de décision

i. Les commandes suivantes permettent de construire l'arbre de décision. Tout d'abord, on doit spécifier quelques paramètres qui précisent comment l'arbre de décision doit être construit. On tape la commande suivante :

```
ad.tennis.cnt<- rpart.control (minsplit = 1)
```

La variable ad.tennis.cnt stocke les paramètres de l'algorithme.

minsplit = 1 signifie que le nombre minimal d'exemples nécessaires à la création d'un nœud est 1. La valeur par défaut est 20. Comme le jeu de données contient moins de 20 exemples, utiliser la valeur par défaut ne produirait pas d'arbre du tout, juste une racine !

Le nom utilisé pour cette variable, ad.tennis.cnt suit la convention R : il indique qu'il s'agît d'un arbre de décision (préfixe ad), pour le jeu de tennis (tennis !) et qu'il s'agît des paramètres de contrôle (cnt) ; des points (.) séparent ces différentes informations. Tout cela est complétement libre ; on aurait pu l'appeler toto, R ne on l'aurait pas interdit. Par contre, pour on, humains, c'est autrement plus parlant ainsi que toto. Vous prendrez donc l'habitude de nommer les variables en suivant ce principe.)

ii. On va construire l'arbre de décision en indiquant :

- l'attribut qui représente la variable cible à prédire : '**la classe : Jouer'**
- les attributs qui doivent être utilisés pour effectuer cette prédiction (pour l'instant, ce seront les 4 autres attributs : **Ciel**, **Température**, **Humidité**et**Vent**)
- l'entrepôt de données avec lequel on construit l'arbre : **Tennis**
- le nom de la variable qui contient les paramètres : **control = ad.tennis.cnt**

```
ad.tennis<-rpart(Jouer~Ciel+Température+Humidité+Vent,tennis,control=ad.tennis.cnt)
```

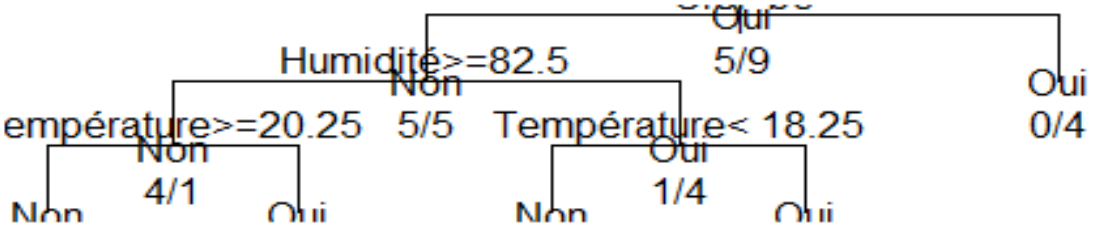
La représentation graphique des Arbres de Décision avec R a deux formes : Représentation Textuelle et Représentation Graphique.

Concernant le premier type de visualisation, l'arbre est donné sous forme de lignes imbriquées dont chacune correspondant à une classe séparatrice. R distingue une variable séparatrice majoritaire (nœud feuille) des classes non majoritaires par le caractère « * »

iii. Afficher le résultat de la construction sous forme de texte:

```
1) root 14 5 Oui (0.3571429 0.6428571)
  2) Ciel=Ensoleillé,Pluie 10 5 Non (0.5000000 0.5000000)
    4) Humidité>=82.5 5 1 Non (0.8000000 0.2000000)
      8) Température>=20.25 4 0 Non (1.0000000 0.0000000) *
      9) Température< 20.25 1 0 Oui (0.0000000 1.0000000) *
    5) Humidité< 82.5 5 1 Oui (0.2000000 0.8000000)
     10) Température< 18.25 1 0 Non (1.0000000 0.0000000) *
     11) Température>=18.25 4 0 Oui (0.0000000 1.0000000) *
  3) Ciel=Couvert 4 0 Oui (0.0000000 1.0000000) *
```

iv. Afficher le résultat de la construction sous forme graphique :
NB : on utilisera les deux commandes ***plot*** et ***text***



v. L'aspect graphique de l'arbre peut être paramétré. Essayer chaque commande et toutes les valeurs affichées :

- *plot (ad.tennis, uniform=T); text (ad.tennis, use.n=T, all=T)*
- *plot (ad.tennis, branch=0); plot (ad.tennis, branch=.7); text (ad.tennis, use.n=T)*
- *plot (ad.tennis, branch=.4, uniform=T, compress=T); text (ad.tennis, all=T,use.n=T)*
- *plot (ad.tennis, branch=.2, uniform=T, compress=T, margin=.1); text (ad.tennis, all=T, use.n=T, fancy=T)*

4. La prédiction de la classe d'une donnée par un arbre de décision

i. La fonction **predict()** utilise un arbre de décision pour prédire la classe de nouvelles données. Elle prend en paramètres l'arbre et un data frame qui contient les données dont il faut prédire la classe. Pour prédire la classe des données du jeu d'exemples (avec lesquels on a construit l'arbre de décision), on tapera la commande :

```
predict(ad.tennis,tennis2)
```

ii. Utilisez l'arbre pour donner une prédiction pour l'entrepôt de données « Tennis2.txt »

```
> predict(ad.tennis,tennis2)
  Non Oui
1   1   0
2   0   1
3   0   1
>
```

II. Construction d'un Arbre de Décision pour le « Cancer du Sein » avec R

Source : http://eric.univ-lyon2.fr/%7Ericco/tanagra/fichiers/breast.txt

Objectifs :

✔ Construire un Arbre de Décision à partir d'un échantillon d'apprentissage contenant 399 observations concernant des patients portant des tumeurs de seins, dans le but de prédire son type : Bénigne ou Maligne.

✔ Appliquer l'arbre construit sur un échantillon de test contenant 300 observations pour spécifier le type tumeur.

1. Préparation des données

[data = breast.app] - Construction de l'arbre sur l'échantillon « breast.app ».
[classe ~.] - Prédire « classe » à partir des autres variables de la base.
[method = « class »] - On construit un arbre de décision c.-à-d. apprentissage supervisé.

```
rm(list=ls())

#chargement des package necessaires

library(rpart)

#chargement de l'échantillon d'apprentissage

breast.app <- read_excel(file.choose())

 #chargement de l'échantillon test

breast.test <- breast.app[,-10]

#appel du package 'rpart' pour l'induction des arbres

ad.tennis.cnt<- rpart.control (minsplit = 1)
```

2. Construction de l'arbre de décision sur l'échantillon breast.app

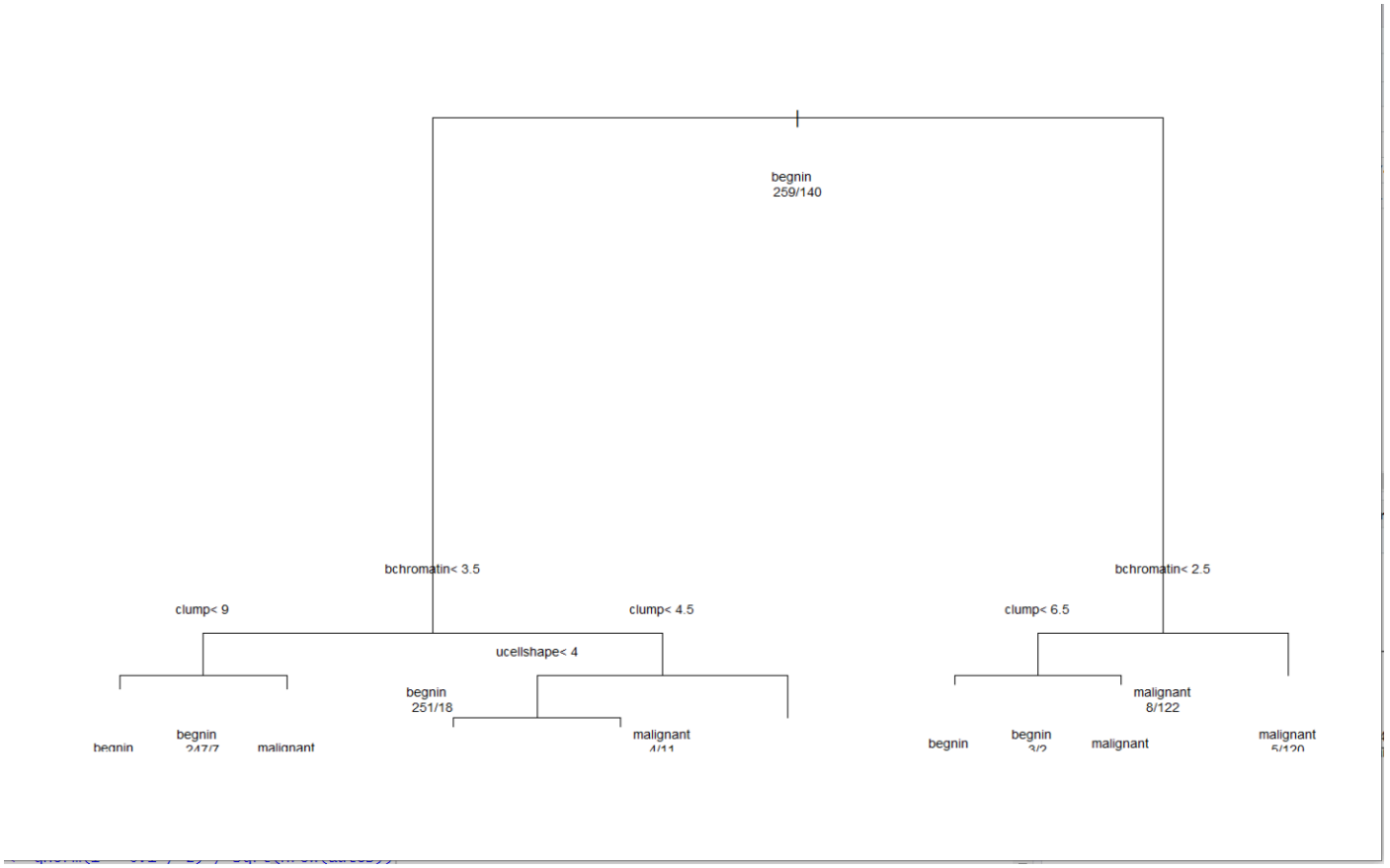
Donner sa description textuelle ci-dessous

```
#arbre de décision avec paramètres par défaut

ad.breast.cnt<- rpart.control (minsplit = 1)

ad.breast <- rpart(classe ~ clump + ucellsize + ucellshape + mgadhesion + sepics +bnuclei + bchromatin + normnucl + mitoses  , data =
breast.app, control = ad.breast.cnt)
```

Donner ci-dessous sa description graphique :



3. Evaluation de l'arbre sur un échantillon test

```
#prédiction sur l'échantillon test

pred.classe<- predict(arbre.1,newdata=breast.test,type="class")

print(summary(pred.classe))

plot(pred.classe)
```

4. Matrice de de Confusion (Table de Contingence)

Cette matrice construit un tableau croisé entre la cible observée (classe) et la prédiction du modèle (pred.classe)

La table mc se comporte comme une matrice à 2 dimensions, on en déduit le taux d'erreur

```
#matrice de confusion

mc <- table(breast.test$classe,pred.classe)

print(mc)

plot(mc)
```

En déduire les pourcentages suivants :

Les Classes de Prédiction		
	Bénigne	Maligne
Bégnine	254	5
Maligne	4	136

5. Calcul du taux de l'erreur :

Calculer le taux d'erreur en appliquant la formule suivante :

erreur = Somme des éléments hors diagonale principale / Nombre total des observations.

```
#taux d'erreur

erreur <- (mc[2,1]+mc[1,2])/sum(mc)

print(erreur)
```

Interpréter les résultats obtenus.
E: représente la proportion d'observations mal classées par rapport au total des observations dans votre échantillon de test.
En d'autres termes, environ 9% des prédictions du modèle ne correspondent pas aux vraies classes.

Cela signifie que le modèle a une performance globalement bonne, car le taux d'erreur est relativement bas.